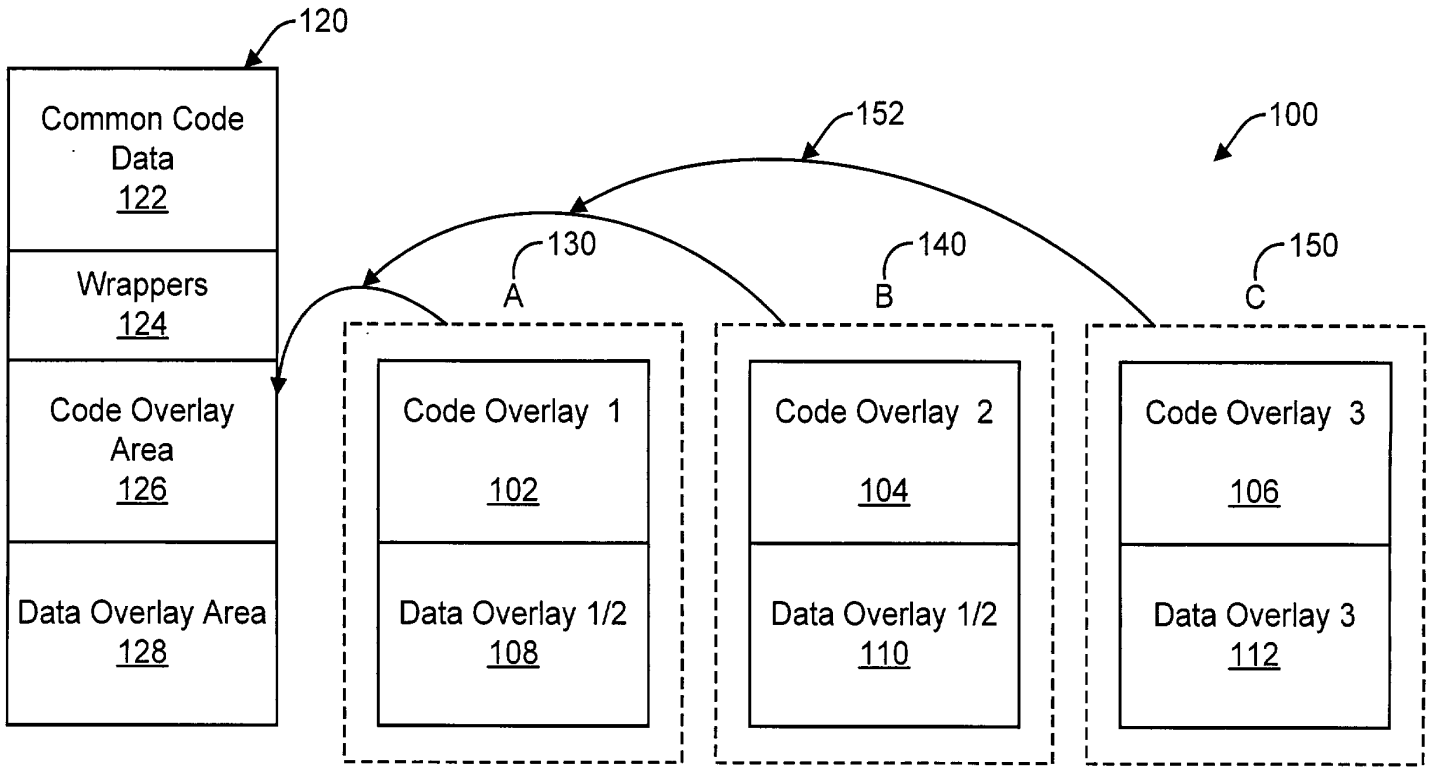

FIG. 1

10032667.04.1802

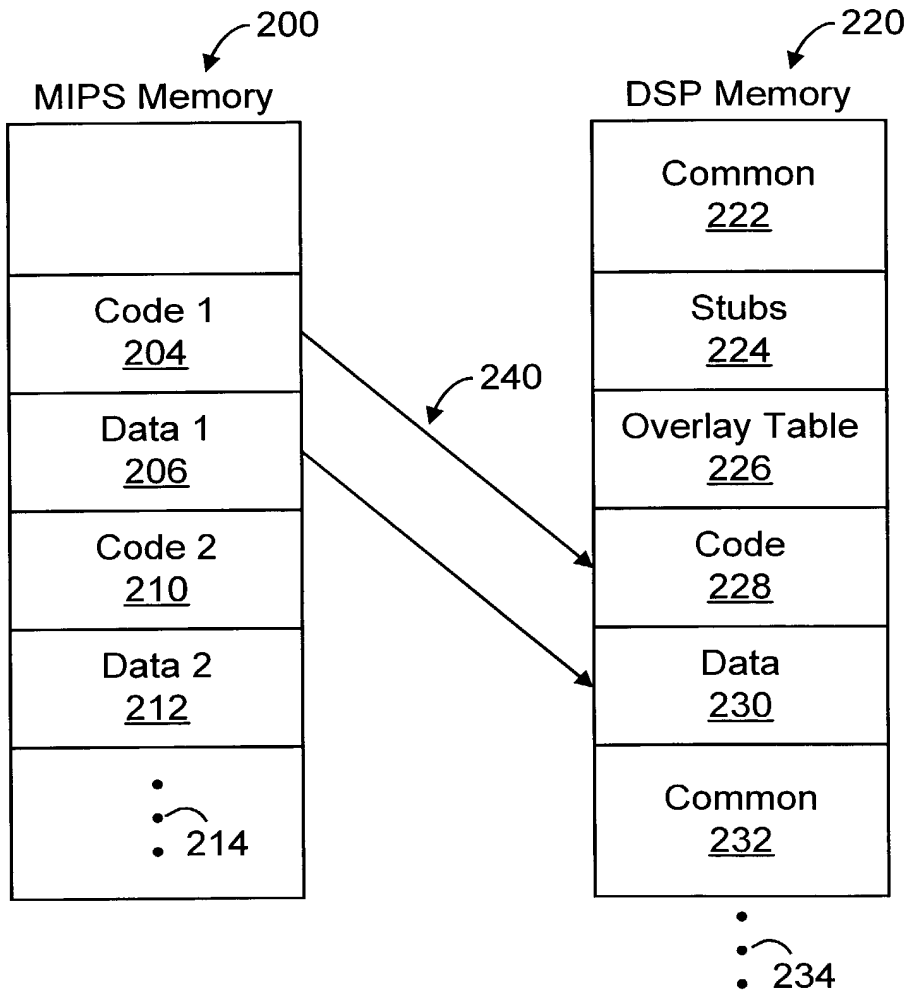
200
MIPS Memory
Code 1
204
Data 1
206
Code 2
210
Data 2
212
214
240
220
DSP Memory
Common
222
Stubs
224
Overlay Table
226
Code
228
Data
230
Common
232
234

FIG. 2

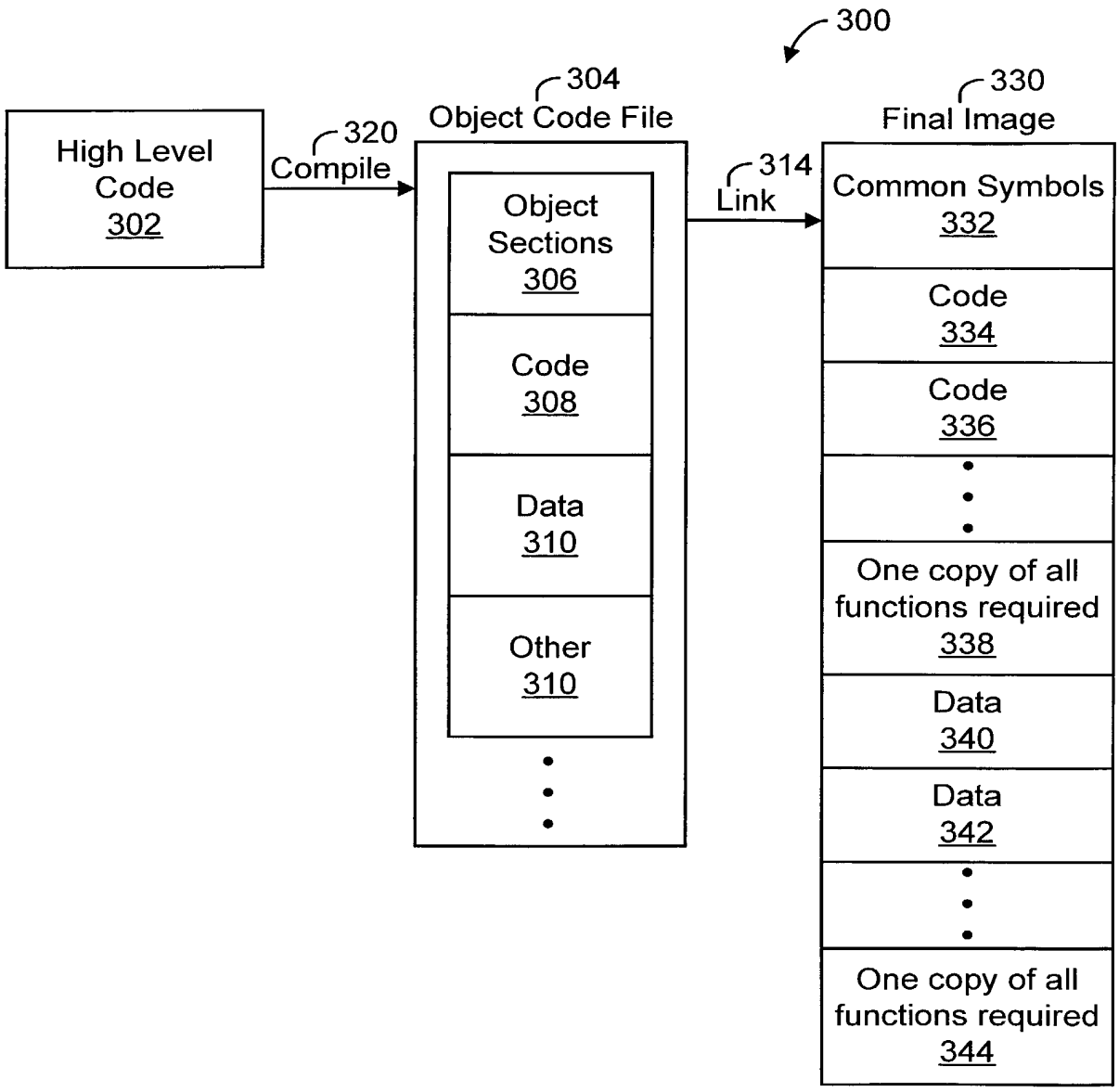
300
304
Object Code File
330
Final Image
High Level Code 302
320
Compile
Object Sections 306
Code 308
Data 310
Other 310
314
Link
Common Symbols 332
Code 334
Code 336
One copy of all functions required 338
Data 340
Data 342
One copy of all functions required 344

FIG. 3A

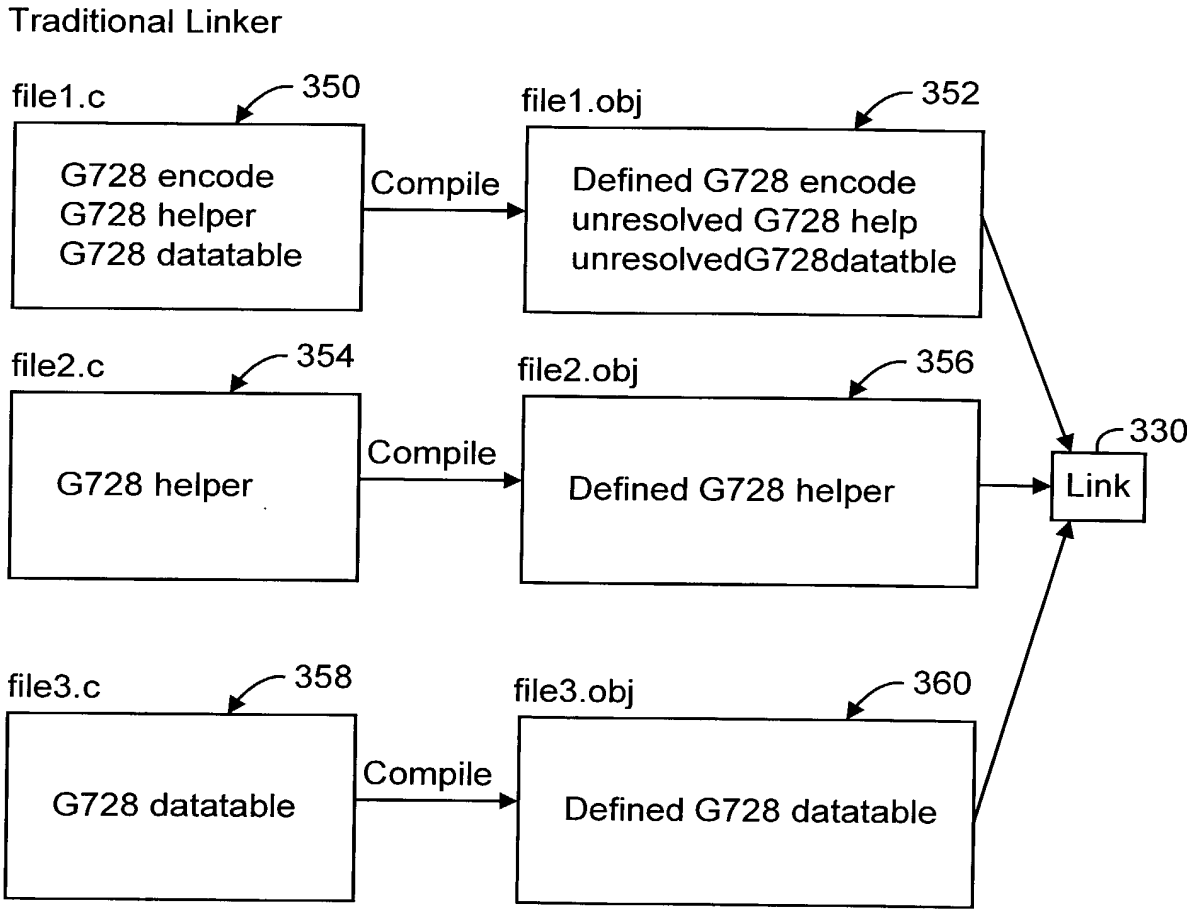
Traditional Linker
file1.c
350
G728 encode
G728 helper
G728 datatable
Compile
file1.obj
352
Defined G728 encode
unresolved G728 help
unresolvedG728datatble
file2.c
354
G728 helper
Compile
file2.obj
356
Defined G728 helper
330
Link
file3.c
358
G728 datatable
Compile
file3.obj
360
Defined G728 datatable

FIG. 3B

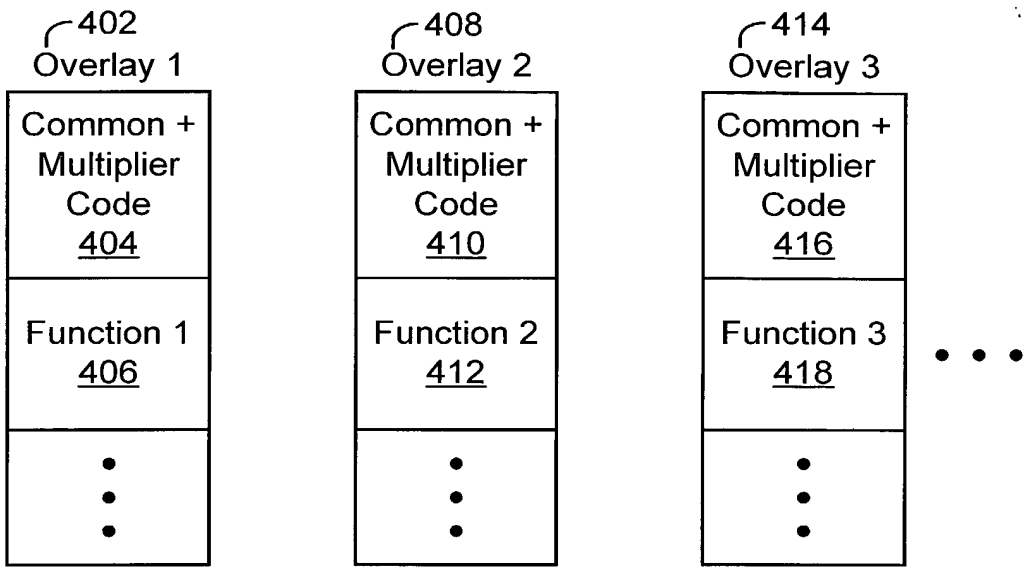
402
Overlay 1
Common + Multiplier Code 404
Function 1 406
408
Overlay 2
Common + Multiplier Code 410
Function 2 412
414
Overlay 3
Common + Multiplier Code 416
Function 3 418

FIG. 4A

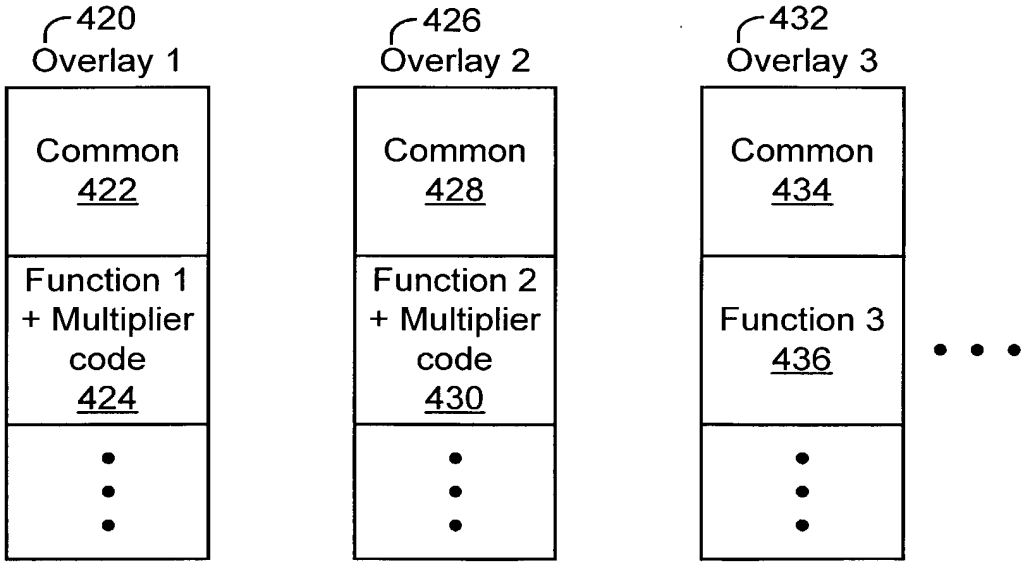
420
Overlay 1
Common 422
Function 1 + Multiplier code 424
426
Overlay 2
Common 428
Function 2 + Multiplier code 430
432
Overlay 3
Common 434
Function 3 436

FIG. 4B

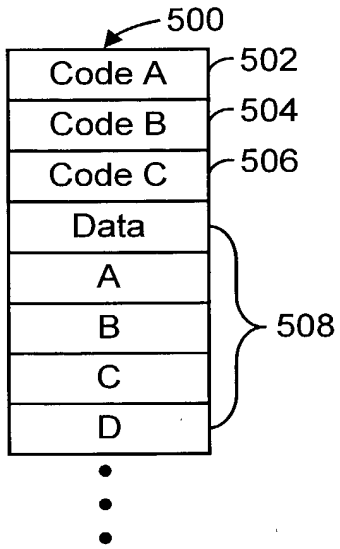
500
Code A
Code B
Code C
Data
A
B
C
D
502
504
506
508

FIG. 5A

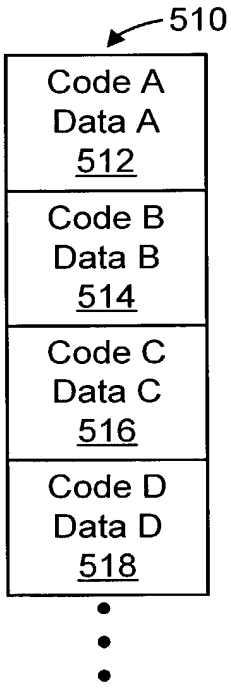
510
Code A
Data A
512
Code B
Data B
514
Code C
Data C
516
Code D
Data D
518

FIG. 5B

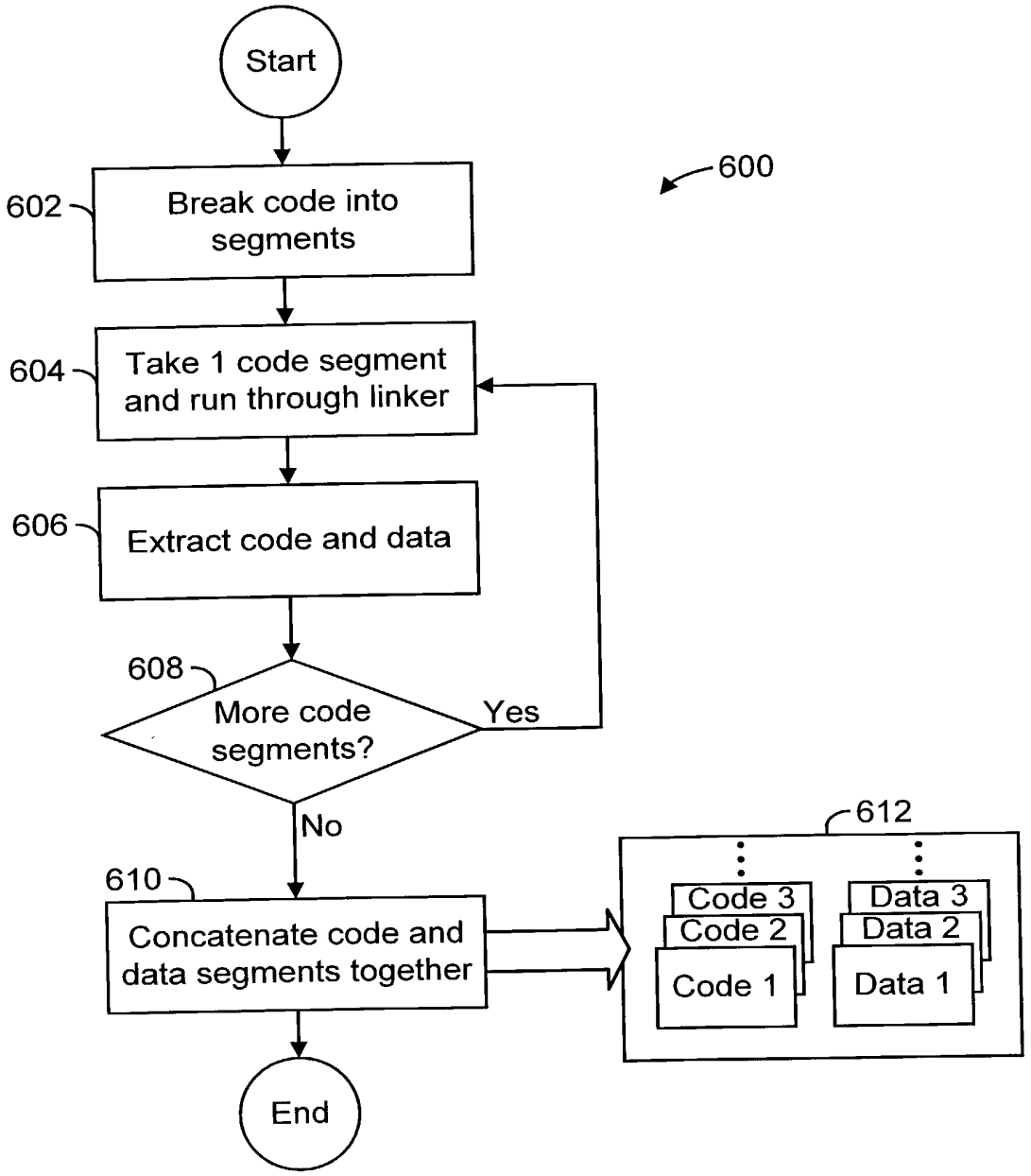
Start
600
602
Break code into segments
604
Take 1 code segment and run through linker
606
Extract code and data
608
More code segments?
Yes
No
610
Concatenate code and data segments together
612
Code 3
Code 2
Code 1
Data 3
Data 2
Data 1
End

FIG. 6A

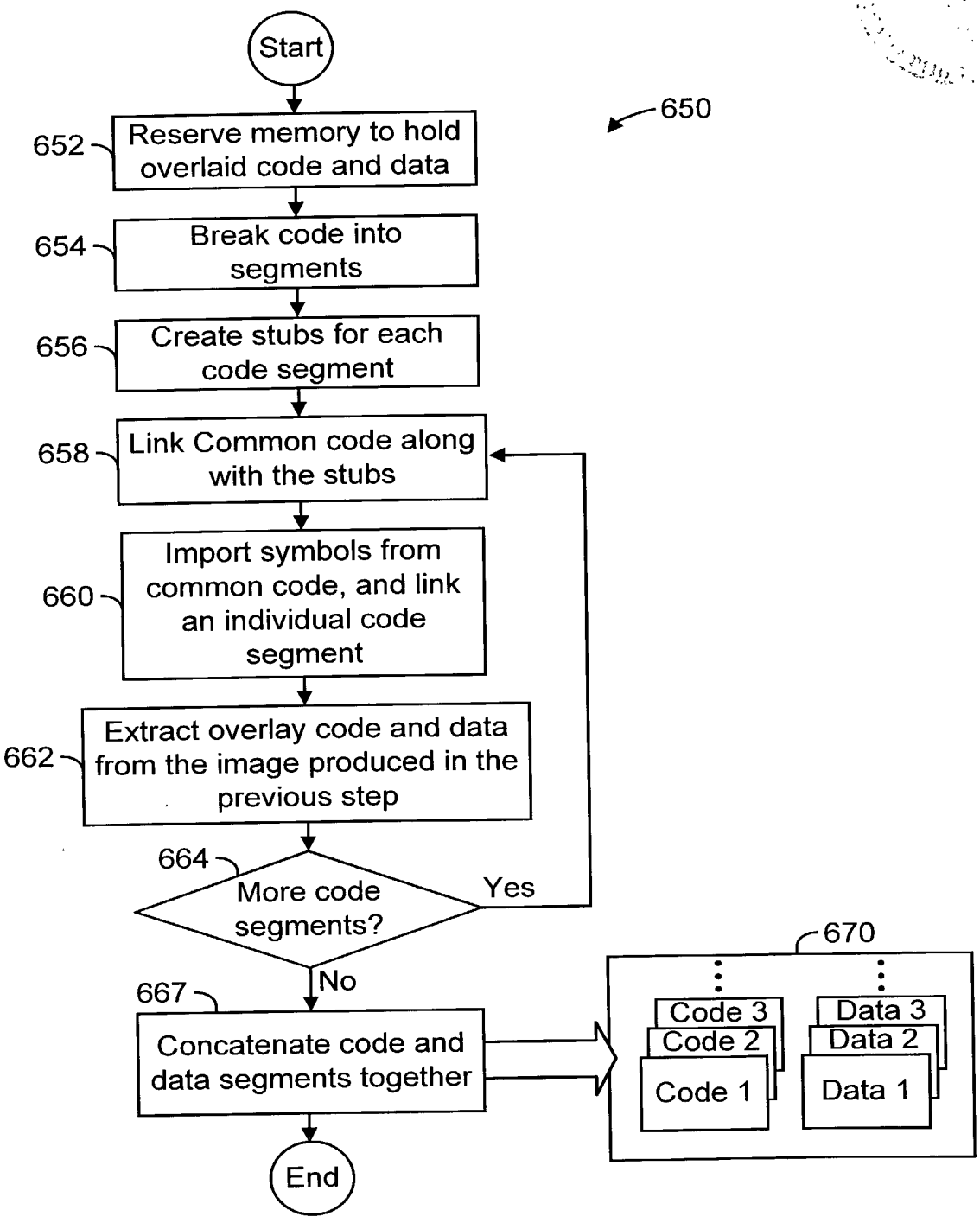
Start
650
652
Reserve memory to hold overlaid code and data
654
Break code into segments
656
Create stubs for each code segment
658
Link Common code along with the stubs
660
Import symbols from common code, and link an individual code segment
662
Extract overlay code and data from the image produced in the previous step
664
More code segments?
Yes
No
667
Concatenate code and data segments together
End
670
Code 3
Code 2
Code 1
Data 3
Data 2
Data 1

FIG. 6B

Detailed Implementation

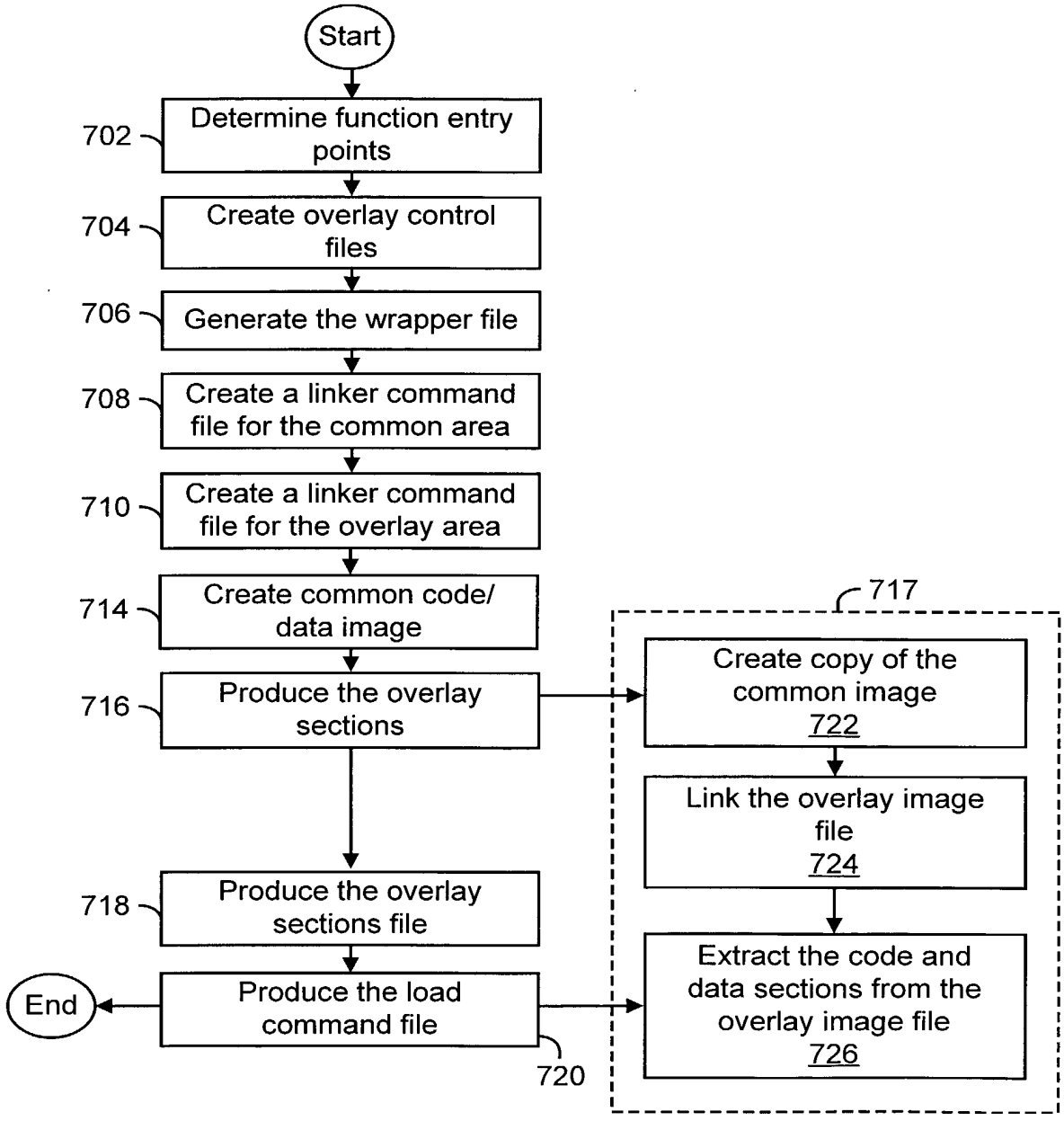

FIG. 7

Linker Command File: Common Area

```
MEMORY
{
    . . . other memory sections removed . . .
    PAGEDCODE(X) : ORIGIN = 0X4200, LENGTH = 0X1800
    PAGEDDATA    : ORIGIN = 0X6000, LENGTH = 0X1400
}
SECTIONS
{
    . . . other sections removed . . .
    pagedCodeSection :
    {
      . = ALIGN (4) :
        _dem_gCodeOverlayAreaStart = ABSOLUTE ( . ) :
      . # . + 0x1800 :
        _dem_gCodeOverlayAreaEnd = ABSOLUTE ( . ) :
    } > PAGEDCODE
    pagedDataSect :
    {
      . = ALIGN (4) :
        _dem_gDataOverlayAreaStart = ABSOLUTE ( . ) :
      . # . + 0x1400 :
        _dem_gDataOverlayAreaEnd = ABSOLUTE ( . ) :
    } > PAGEDDATA
    .overlayDescr :
    {   _dem_gOverlayDescr = ABSOLUTE ( . ) :
      SHORT ( -2 )
    } > INTDATA
}
```

FIG. 8A

Overlay Area

```
MEMORY
{
    . . . other memory sections removed . . .
    PAGEDCODE(X) : ORIGIN = 0X4200, LENGTH = 0X1800
    PAGEDDATA    : ORIGIN = 0X6000, LENGTH = 0X1400
}
SECTIONS
{
    . . . other sections removed . . .
    .text :
    {
      . = Align (4) :
      * ( . text)
    } > PAGEDCODE
    .data :
    {   . = Align (4) :

    }   } > PAGEDDATA
```

FIG. 8B